

REMARKS

A. Background

Claims 1-19 were pending in the application at the time of the Office Action. Claims 1-19 were rejected as being non-statutory subject matter under 35 U.S.C. 101. Claims 1-19 were also rejected as being anticipated over the prior art. By this response, Applicant has cancelled claims 17-19 and added new claims 20-33. As such, claims 1-16 and 20-33 are presented for the Examiner's consideration in light of the following remarks.

B. Examiner Telephone Interview

Applicant(s) and applicant's attorney express appreciation to the Examiner for the courtesies extended during the recent interview held on September 12, 2007. This response includes the substance of the Interview.

C. Proposed Amendments

By this amendment, claims 17-19 have been cancelled without prejudice or disclaimer and claims 20-33 have been added. Applicant asserts that the addition of new claims 20-33 are based on the specification as originally filed and that no new matter has been added. As such, Applicant requests that the addition of new claims 20-33 be entered.

D. Objections

Page 2 of the Office Action objected to claim 2 based on an informality. In addition, the Office Action recommended amending claim 2 to define the system with steps or acts beginning with an active verb. By this response, Applicant has amended claim 2 to correct the informality. Applicant has also amended claim 2 to recite "a communication module". As such, Applicant respectfully requests that the objection to claim 2 be withdrawn.

E. Rejection on the Merits

Non-statutory Subject Matter under 35 U.S.C. 101

Claims 1 and 17 were rejected under 35 U.S.C. 101 as being non-statutory subject matter. By this response, claim 17 has been cancelled and, as such, Applicant respectfully submits that the rejection with respect to claim 17 is now moot.

In addition, Applicant asserts that claim 1 presents statutory subject matter under 35 U.S.C. 101 since it is clearly based in a computer-implemented environment. However, in order to advance allowance of claim 1, Applicant has amended claim 1 to recite "a set of management and design tools stored on physical computer-readable media that, when executed by the server infrastructure, causes the system to perform management and development of software modules as services."

Applicant respectfully submits that claim 1 and dependent claims 2-16 thus present patentable subject matter.

Indefiniteness under 35 U.S.C. 112

Page 3 of the Office Action rejected claim 1-19 as being indefinite under 35 U.S.C. 112, second paragraph, pointing various instances where the Examiner believed this to be the case. Applicant notes that claims 17-19 have been cancelled and so the indefiniteness rejection with respect to these claims is render moot.

With respect to claim 1-16, Applicant has made amendments to address the portions that the Examiner believed were indefinite in order to advance allowance of the claims. As such, Applicant respectfully requests that the indefiniteness rejection to claims 1-16 be withdrawn.

Anticipation under 35 U.S.C. 102

Claims 1-3, 5-15, 17 and 18 were rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Pat. Pub. No. 2005/0044197 A1 to Lai. Applicant has cancelled claims 17 and 18, rendering the anticipation rejection with respect to these claims as moot.

As discussed in the telephone interview, and as clarified in the claims, the present invention is related to a software development system for developing service modules. The unique aspect of the invention is that the software development system that is used to generate service modules *is itself also built on service modules*. In other words, the software development

system leverages *its own service-oriented artifacts* to perform its own system functionality. See Specification, [para 3].

Once it is understood that the software development system itself is built upon at least some service modules, it becomes clear that the cited references do not teach this unique concept. For example, the main Lai reference teaches:

[0059] Embodiments of a system and method for providing a generic, vendor-independent Web Services architecture incorporating a structured methodology and design patterns for designing and implementing Web Services are described. Embodiments may incorporate a structured methodology, best practices and design patterns that address reliability, availability and scalability of Web Services architecture. Embodiments may provide a mechanism for designing and implementing Web Services as business (or other application) solutions that may include mainframe and legacy systems interoperability and cross-enterprise integration. Embodiments may provide mechanisms for integrating heterogeneous technology components into Web Services solutions. Embodiments may provide a vendor-independent Web Services architecture framework and reusable Web Services design patterns, which may help in creating end-to-end solutions based on past experience and best practices. Embodiments may include best practices for delivering Web Services solutions with Quality of Services.

....

[0063] In one embodiment, a Web Service includes a service provider configured to provide one or more services and one or more service requesters configured to access the one or more services from the service provider via a network. In one embodiment, the Web Service is a Business-to-Consumer Web Service, the service provider is a business service provider, and the service requester is an end user. In one embodiment, the Web Service is a Business-to-Business Web Service, the service provider is a business service provider, and the service requester is a server. In one embodiment, the Web Service includes a service broker configured to interact with the service provider and service requester to negotiate and provide the services of the service provider to the service requester. In one embodiment, the service provider may act as a service broker. One embodiment may include a service registry. The service provider may be configured to register and publish the services in the service registry, and the service requester may be configured to discover the service provider through the service registry.

‘197 application, para. 59, 63. Assuming that a service provider is similar to the service-oriented software development system as recited in independent claims 1 and 20, there is no teaching in Lai that the service provider itself is built upon at least some service modules.

Thus, Lai does not teach "wherein the system uses software service modules, implemented through the server infrastructure itself and developed and managed based on functionality of the set of management and design tools provided to end-users of the system, to implement its own software functionality" as recited in claim 1. As such, Applicant respectfully requests that the anticipation rejection with respect to claim 1 be withdrawn.

Dependent claims 2-16 depend from independent claim 1 and thus incorporate the limitations thereof. As such, Applicant respectfully submits that claims 2-16 are distinguishable over the prior art for at least the same reasons discussed above with respect to claim 1 and request that the anticipation rejection with respect to claim 1 be withdrawn.

Obviousness Rejections under 35 U.S.C. 102

Pages 10 through 12 of the office action rejected claim 4 as obvious over Lai in view of U.S. Pat. Pub. No. 2004/0015564 to Williams and claim 16 as obvious over Lai in view of U.S. Pat. Pub. No. 2005/0015491 A1 to Koeppel. However, neither Williams nor Koeppel make up for the deficiencies of Lai. As such, Applicant respectfully requests that the obviousness rejection with respect to claims 4 and 16 be withdrawn.

F. New Claims

New independent claims 20 and 27 are distinguishable over the cited reference. For much the same reason discussed above, Lai, Williams and/or Koeppel do not teach "each of the user interface tool and run-time engine comprising system functions to enable the operation of the service-oriented development system, at least some of the system functions of the user interface tool and run-time engine *being themselves built to use service modules* that are developed using the user interface tool and run-time engine, wherein the core module is configured to service the at least some of the system functions that are built as service modules" as recited in new claim 20.

Finally, Lai teaches, with reference to Figure 25:

[0547] The SOAP client may then generate a SOAP request in an XML document and send it to the SOAP server (RPC router servlet). The SOAP server may initiate an RPC call to the application providing the business services. Using a synchronous Web Services call, an acknowledgement and the inquiry details (or

transaction information) may be returned to the SOAP client. Both the caching event and the business transaction may be logged for tracking and management purposes.

....

[0640] FIG. 37 is a SOAP Logger Sequence Diagram according to one embodiment, and elaborates on the above Use Cases. Once the SOAP client initiates a SOAP service request, the SOAP server may unmarshal the sender information from the SOAP envelope and the session information. The SOAP server logs the sender and session information in the logger using the Java API for logging. The SOAP server may then invoke RPC calls with the remote business applications. The date/time and usage may also be logged in the logger. Upon successful invocation of the remote business services, the SOAP server may acknowledge with a return code or an exception. The information captured in the logger may be sufficient for basic usage analysis and per-call-based billing.

....

[1202] Each transactional Web Services call is preferably logged at the level of Web Services invocation and transport layer. This is in addition to the transaction log taken by the local or remote applications. In such a case, administrators may track and trace the service request at different points within the life cycle. For example, the HTTP/S activities may be tracked from the Web server's audit and log, and the XML-RPC SOAP calls may be traced from the SOAP server log.

‘197 application (emphasis added). Lai specifies that a SOAP client generates a SOAP request, so it is clear that the type of invocation is known to the consumer. Thus, Lai does not teach that invocation of the service is encapsulated from the consumer and, in fact, does not discuss what happens at the consumer end when the consumer makes an invocation request. As such, Lai does not teach “wherein operation of the internal invoker and remote invoker are encapsulated from the implementer by the invoker interface such that the consumer is not aware whether the invocation request is being sent via the local invoker or the remote invoker” as recited in new claim 27.

Thus, new claims 20 and 27 and their dependent claims are distinguishable over the cited references.

G. Conclusion

Applicant notes that this response does not discuss every reason why the presented claims are distinguished over the cited prior art. Most notably, applicant submits that many if not all of the dependent claims are independently distinguishable over the cited prior art. Applicant has

merely submitted those arguments which it considers sufficient to clearly distinguish the claims over the cited prior art.

In view of the foregoing, applicant respectfully requests the Examiner's consideration and allowance of 1-16 and 20-33 as presented herein.

In the event there remains any impediment to allowance of the claims which could be clarified in a telephonic interview, the Examiner is respectfully requested to initiate such an interview with the undersigned.

Dated this 26th day of September 2007.

Respectfully submitted,

/sara d. jones/Reg. # 47,691
SARA D. JONES
Attorney for Applicant
Registration No. 47,691
Customer No.: 022913
Telephone No.: 801-533-9800

SDJ:vlr
SDJ0000001518V001